

MÉTODOS DE ENSEÑANZA EN INGENIERÍA DEL SOFTWARE: HACIA UNA COOPERACIÓN UNIVERSIDAD-EMPRESA.

Jaime Gómez, Rafael Muñoz
Departamento de Lenguajes y Sistemas Informáticos.
Universidad de Alicante.
e-mail: {jaime,rafael}@dlsi.ua.es

RESUMEN

Enseñar a resolver los problemas que aparecen desde las primeras etapas de la construcción de software es una tarea compleja, fundamentalmente porque el análisis y diseño de sistemas de información depende, en gran medida, de situaciones que son difíciles de recrear en un aula. El objetivo de este artículo es presentar la experiencias adquiridas en la aplicación de un método de enseñanza basado en la interacción alumno-empresa a través del cual introducimos al alumno en el mundo real enfrentándole ante un problema de sistemas de información que debe de resolver. Este método de trabajo que hemos utilizado durante los últimos tres años, nos ha permitido comprobar que el alumno resuelve los problemas que aparecen en las primeras etapas del ciclo de vida del software de forma más fluida y eficiente.

1. OBJETIVOS

La Ingeniería del Software en el Departamento Lenguajes y Sistemas Informáticos de la Universidad de Alicante se encuentra dividida en dos asignaturas que se imparten en el tercer año de las carreras de informática (Ing. Técnica de Gestión, Ingeniería Técnica de Sistemas e Ingeniería Informática), pero en diferentes cuatrimestres. En el primer cuatrimestre se imparte la asignatura de Análisis y Especificación de Sistemas de Información, y en el segundo cuatrimestre la asignatura de Ingeniería del Software I.

Aunque cada una de ellas tiene sus objetivos concretos, podemos destacar como objetivos generales de éstas los siguientes:

- Concienciar al alumno que uno de los parámetros significativos a tener en cuenta a la hora de idear una estrategia para el correcto desarrollo de un proyecto software es el tamaño del problema.
- Diferenciar las etapas o fases que constituyen lo que se conoce como ciclo de vida del software.

- Aprender los diferentes métodos y técnicas que resuelven los problemas que se presentan en cada una de estas fases.
- Saber aplicar los métodos de manera eficiente sobre un problema concreto para obtener la mejor solución.

Estos objetivos generales se concretan dependiendo de la etapa del ciclo de vida que estemos abordando.

La etapa de **análisis** se estudia en la asignatura “Análisis y Especificación de Sistemas de Información”, sus objetivos principales son:

- Desarrollar la capacidad de examinar un sistema de información y determinar si el empleo de un ordenador es la solución apropiada.
- Saber como recolectar e interpretar hechos que ayuden a diagnosticar un problema organizacional y la forma en la que se relaciona con la computación y los sistemas de información.
- Adquirir soltura para diseñar y desarrollar las especificaciones de un sistema de información a partir del examen del sistema actual.
- Aprender una metodología de análisis estructurado para modelizar los componentes de un sistema de información.

Por otro lado, la etapa de **diseño** se estudia en la asignatura de “Ingeniería del Software I”, y sus objetivos principales son:

- Utilizar la abstracción y el diseño modular como mecanismos de ayuda para simplificar y reutilizar los componentes del software.
- Aprender a construir los modelos del diseño estructurado a partir de los modelos del análisis.
- Saber evaluar la calidad del diseño global de un sistema en función de las características de sus módulos.
- Conocer los fundamentos del diseño orientado a objetos como alternativa importante en el diseño de software.

Junto con estas dos asignaturas la docencia en Ingeniería del Software en nuestra Universidad se completa con otra asignatura más “Ingeniería de Software II” que es impartida por otro departamento en un curso posterior.

2. CONTENIDOS QUE SE IMPARTEN

Los contenidos de las asignaturas se han diseñado teniendo en cuenta los objetivos marcados para cada una de ellas. Esto nos ha llevado a intentar respetar al máximo tanto el orden de presentación como la planificación temporal (nº de horas), de los temas seleccionados puesto que somos varios los profesores que impartimos estas asignaturas.

Análisis y Especificación y Sistemas de Información: Sus contenidos teóricos se encuentran agrupados en dos módulos (ver tabla 1); un módulo de introducción al desarrollo de sistemas de información donde se enseñan fundamentalmente técnica de análisis y determinación de requerimientos software según [4, 5, 8] y otro módulo donde se enseña una metodología de análisis de software que en nuestro caso se basa en el paradigma estructurado según [9]:

| |
|--|
| I. Introducción al desarrollo de sistemas de información. |
| T.1. Introducción al análisis de sistemas. |
| T.2. Análisis y determinación de requerimientos. |
| II. Estrategia de desarrollo por análisis estructurado. |
| T.3. Análisis estructurado. |
| T.4. Diagramas de flujos de datos. |
| T.5. Diccionario de datos. |
| T.6. Especificaciones de procesos. |
| T.7. Diagramas entidad/relación. |
| T.8. Consistencia entre modelos. |
| T.9. Diagrama de transición de estados. |
| T.10. Casos de estudio. |

Tabla 1: Contenidos de Análisis y especificación de sistemas de información

Ingeniería del Software I: Los contenidos teóricos de esta asignatura como se puede ver en la tabla 2 se encuentran agrupados en tres módulos; un primer módulo de introducción sobre los comienzos de la Ingeniería del software [7], un segundo módulo donde se enseña una metodología de diseño basada en el paradigma estructurado [6] y finalmente un tercer módulo donde se presenta una metodología de diseño basada en el paradigma objetual según [1]:

| |
|---|
| I. Introducción. |
| T.1. Introducción a la Ingeniería del software. |
| II. Diseño estructurado. |
| T.2. Introducción al diseño. El diagrama de estructura. |
| T.3. Métodos de especificación de módulos. |
| T.4. Acoplamiento. |
| T.5. Cohesión. |
| T.6. Guías adicionales de diseño. |
| T.7. Derivar el diagrama de estructura. |
| T.8. Casos de estudio. |
| III. Metodología orientada al objeto |
| T.9. Introducción al paradigma objetual. |
| T.10. Metodología de un diseño orientado a objeto. |
| T.11. Proceso de un diseño orientado a objeto. |
| T.12. Caso de estudio. |

Tabla 2: Contenidos de Ingeniería del software I

3. MÉTODO DE ENSEÑANZA

El método de enseñanza que utilizamos consiste fundamentalmente en la aplicación de los conceptos teóricos que se aprenden sobre casos reales. Para ello, introducimos al alumno en un entorno preciso donde pueda enfrentarse con los problemas que debe de resolver.

Esta incursión en el mundo real, es posible gracias a convenios de colaboración con empresas de diferentes ámbitos de nuestra provincia que permiten a los alumnos realizar períodos de presencia física breves (aproximadamente 15 días), en los cuales trabajan sobre las tareas asociadas al análisis de un determinado sistema de información dentro de ella.

El motivo que nos ha llevado a esta propuesta han sido las fuertes críticas [2, 3], que a sufrido la clase magistral por parte de pedagogos y universitarios, a partir de la convicción de que ésta constituye un método de transmisión de conocimientos esencialmente dogmático, ya que no induce en el receptor el aprendizaje de habilidades y establece, en general, una barrera entre el conocimiento y la aplicación del mismo.

Sin embargo, pensamos que la existencia de este rotundo y generalizado rechazo hacia la clase magistral es debido más por la existencia de cierto estereotipo negativo de la misma, que por el análisis detenido y sereno de las posibilidades docentes del modelo en sí, que son múltiples. Pensamos que la clase magistral no tiene que ser unidireccional entre el profesor que expone y el alumno que recibe pasivamente la información. Sino que se puede hacer una clase que invite al diálogo y a la discusión de diferentes aspectos de un problema, lo cual nos lleva a una relación mas estrecha con los alumnos.

Esta relación se ve enriquecida gracias a la motivación que tienen los alumnos debido a que van a enfrentarse a problemas reales, cuestión que les hace ver una aplicación práctica inmediata. Para ello, necesitan comprender todos los aspectos teóricos que se presentan en las clases magistrales. Durante los años que venimos poniendo en práctica este método hemos observado como los alumnos participan activamente en estas clases haciendo todo tipo de preguntas, e incluso proponiendo diferentes problemas.

La formación del alumno se completa con las clases de problemas y las clases prácticas en el laboratorio.

Las clases de problemas constituyen una prolongación de las clases magistrales. Estas clases nos sirven para contrastar la solidez de los conocimientos teóricos y si estos ha sido comprendidos o memorizados.

Durante las clases magistrales proponemos a modo de ejercicio la resolución de cuestiones abiertas o la mejora de soluciones dadas, las cuales se resolverán en estas clases de problemas como parte del trabajo a realizar. No solo se realizan los ejercicios que proponemos, sino que también tienen cabida aquellos problemas planteados por los alumnos en otras sesiones.

La corrección de estos ejercicios se realiza de forma interactiva siendo los alumnos los que propondrán las soluciones que serán comentadas por el resto, actuando nosotros como meros moderadores. De esta forma se les obliga a razonar sobre la marcha, a comparar soluciones y a llevar al día la asignatura. Además esta dinámica les ayudará mejorar su capacidad de expresión. Por otra parte nos sirve para disponer de un mayor número de datos a la hora de evaluar al alumno.

Las prácticas en el laboratorio se realizan en grupos de cuatro personas. Grupos mayores dificultan la evaluación del trabajo realizado por cada estudiante. Aunque realmente lo que marca el número de personas que intervendrán en un grupo será la naturaleza del trabajo. Por lo que si se requiriera de grupos más numerosos, el profesor se encargaría de fragmentar dicho trabajo en partes coordinadas entre sí.

El trabajo a realizar en las clases de laboratorio consiste básicamente en el aprendizaje del manejo de las herramientas a utilizar y la realización de las prácticas que será lo que el profesor evalúe junto con la documentación elaborada del sistema de información escogido.

Como hemos comentado anteriormente, no debemos olvidar que parte de estas clases de laboratorio incluyen un periodo de presencia física, aproximadamente 15 días, en la empresa. Los alumnos tienen un primer contacto con las empresas contemplando la realidad actual e intuyendo las necesidades profesionales del sector. Hemos podido comprobar como esta situación suscita en el estudiante un intento de mejora y ampliación de sus conocimientos en una determinada área.

La información que obtienen los alumnos durante este período les sirve para elaborar la documentación asociada a las prácticas de cada asignatura. En la Análisis y especificación de sistemas de información presentan los siguientes contenidos:

- *Actividad principal de la empresa y sector:* Esta información nos permite comparar distintas empresas del mismo sector, localizando las características comunes que aparecen en ellas.
- *Organigrama jerárquico:* Es interesante disponer de las dependencias funcionales entre los diferentes componentes de una organización, pues nos ayuda a ver los diferentes subsistemas de información de la empresa.
- *Análisis de factibilidad:* Cuando se le asigna un sistema al grupo de trabajo, no sabe que parte de él va a analizar. Es importante realizar un estudio de este tipo para asignar prioridades dentro de los diferentes subsistemas de información.
- *Delimitación del proyecto:* Una vez establecidas las prioridades sobre los diferentes subsistemas, habrá que tomar la decisión de cuál de ellos se analizarán. Esta decisión es tomada en conjunto (profesor y grupo de trabajo).
- *Análisis de requerimientos:* El alumno debe obtener la información clave referente a los subsistemas a desarrollar. Para ello se apoya en el uso de herramientas como: entrevistas, cuestionarios, documentos, formularios, etc.
- *Formalización de requerimientos:* Se generan una serie de plantillas para cada uno de los requerimientos software. Las plantillas tienen el mismo formato para cualquier tipo de requerimiento.
- *Construcción de los modelos de análisis:* En este punto el grupo de trabajo suele dividirse en dos subgrupos, encargándose uno de ellos del modelo de datos del sistema y el otro del modelo de proceso del sistema. Hay que señalar que debe existir un alto grado de coordinación entre ambos subgrupos, para que paralelamente se vaya construyendo entre ambos el diccionario de datos del sistema. Periódicamente durante este proceso se debe proceder a comprobar la consistencia entre modelos.

Esta documentación es la que se considera válida para evaluar la parte práctica de la asignatura. Además, puesto que la mayoría de nuestros alumnos cursan en el segundo cuatrimestre la asignatura Ingeniería del Software I, las prácticas de esta última toman como punto de partida la documentación anterior cuyos contenidos principales son dos:

- *Construcción de los modelos de diseño:* tanto para la versión estructurada como para la versión orientada a objetos. Esto les permite comparar soluciones de diseño alternativas para especificaciones software comunes y observar en que casos es mejor utilizar una u otra.
- *Generación del prototipo:* la generación automática de programas (utilidad que dispone la herramienta Multibase, comentada más adelante), permite al alumno juzgar hasta que punto el prototipo obtenido es funcionalmente equivalente a las especificaciones del análisis del sistema. La parte estática es modelizada sobre un sistema de gestión de base de datos y la parte dinámica sobre programas del lenguaje Ctl.

Con esta planificación conseguimos una integración total de las prácticas realizadas en cada una de las asignaturas y el alumno puede apreciar una continuidad precisa en las tareas de análisis y diseño que en nuestro caso llega hasta la obtención de un producto software inicial.

4. PAPEL DE LA PROGRAMACIÓN Y DE LAS HERRAMIENTAS.

Como se puede apreciar, las herramientas de análisis y diseño tiene un papel más importante que los lenguajes de programación en los trabajos prácticos de las asignaturas, de hecho éstos últimos sólo son utilizados en la etapa de diseño para la personalización del prototipo generado.

En la etapa de análisis, se utiliza como soporte gráfico para la construcción de sus modelos la herramienta uppercase Silver-Run. En la etapa de diseño utilizamos el entorno de desarrollo Multibase y la herramienta Domain como soporte gráfico de los modelos de diseño orientados a objetos.

Realmente, la programación juega un papel secundario ello es debido a que nuestro objetivo es básicamente realizar análisis y diseño. Únicamente, en la segunda de las asignaturas se le propone al alumno la posibilidad de personalizar el prototipo generado mediante la introducción de código adicional, ello requiere al menos un mínimo conocimiento del lenguaje Ctl de Multibase.

5. LOS LENGUAJES DE PROGRAMACIÓN QUE UTILIZAN EN LOS TRABAJOS PRÁCTICOS

El lenguaje Ctl permite introducir cierto grado de implementación en la etapa de diseño. Para ello ponemos a disposición del alumno manuales y resúmenes de uso básico. Además, planificamos seminarios de libre asistencia (2 horas) en los cuales enseñamos algunas de las posibilidades más interesantes del lenguaje que pueden aplicar en sus trabajos.

Durante estos últimos años hemos podido apreciar lo difícil y pesado que se hace manipular, revisar, archivar y calificar los trabajos prácticos que entregan nuestros alumnos. Esto nos ha llevado a proponer la entrega de la documentación en formato "Html" siguiendo las novedosas propuestas que se están produciendo actualmente en el empleo de las Nuevas Tecnologías aplicadas a la enseñanza. El coste para el alumno no es elevado pues además de ofrecerles un seminario de introducción al Html (2 horas) existen infinidad de herramientas que realizan la conversión de información textual a este formato.

Creemos que esta propuesta nos va a ayudar en años venideros a ofertar trabajos mas interesantes como continuación a los ya realizados en cursos anteriores y por consiguiente evitar posibles redundancias en el desarrollo de alguno de ellos.

6. TIPOS DE TRABAJOS PRÁCTICOS QUE SE PLANTEAN A LOS ESTUDIANTES.

Los tipos de trabajos que se ofertan a los alumnos pertenecen fundamentalmente al ámbito empresarial de la provincia. Para ello, proponemos una serie de empresas (a algunas de ellas nos unen convenios de colaboración), que pertenecen a diferentes sectores de nuestra industria tanto públicos (hospitales, bibliotecas, administración,...) como privados (calzado, construcción, juguetes, asesorías,...).

No obstante, damos libertad al alumno para que pueda realizar su trabajo práctico en otra empresa que por cualquier motivo resulte mas interesante para él, siempre y cuando el sistema de información que propone el alumno cumpla con los requerimientos mínimos de tamaño y complejidad exigidos.

Como podemos ver se intenta abarcar todo el ámbito de lo que es la empresa en la provincia de Alicante, que en definitiva serán los tipos de problemas que puedan encontrarse una vez finalizado sus estudios.

7. ESTRATEGIAS PARA INCORPORAR LA PROGRAMACIÓN EN EL CURRÍCULUM DE LA ASIGNATURA

Aunque los planes de estudios marcan que para poder cursar este tipo de asignaturas se tienen que haber aprobado antes otras relacionadas con la disciplina de la programación. No por ello debemos dejar de lado el uso de estos lenguajes y aplicarlos en la enseñanza de la Ingeniería del Software. A nuestro entender, existen dos estrategias básicas para incorporar la programación a las asignaturas que impartimos:

- la primera de ellas es la incorporación del paradigma de programación automática en la etapa de diseño del software.
- la segunda es la propuesta de asignaturas sobre enseñanza en técnicas de reingeniería y rehusos de software.

Esta primera estrategia es la que intentamos seguir cuando proponemos la utilización de un entorno de desarrollo software para obtener un prototipo automático a partir de las especificaciones de diseño obtenidas.

Lamentablemente notamos en falta en nuestra Universidad la existencia de otras asignaturas como las que se proponen en la segunda de las alternativas que nos permitiesen evaluar ciertos aspectos de la construcción del software que son mas cercanos a los lenguajes de programación.

8. MATERIAL DIDÁCTICO UTILIZADO

Antes del comienzo de cada clase el alumno dispone de esquemas, resúmenes y una buena colección de ejemplos, esto le permite centrar su atención en la comprensión de los conceptos y no en la recopilación de ellos. Las exposiciones orales de los temas de la asignatura se ve complementada con la proyección de transparencias y la pizarra.

El uso de las nuevas tecnologías no solo se ve reflejado en la entrega de los trabajos prácticos de nuestros alumnos, sino que también y como novedad de este último año se ha montado una infraestructura de intranet que nos permite disponer de un servidor Web docente en Ingeniería del Software.

A través de este Web, el alumno puede acceder por un lado a la información y las herramientas que necesita para realizar las distintas actividades y por otro a una serie de servicios añadidos (consulta de problemas, revisiones parciales, entrega normalizada de documentación, ...) que le facilita la realización de los trabajos a presentar. Las distintas funcionalidades del servidor se han diseñado durante este último año y continuarán durante el que viene.

9. CONCLUSIONES

La experiencia adquirida en la aplicación de este método durante los tres últimos años nos hace reflexionar sobre ciertas características de la enseñanza en Ingeniería del Software:

- es lo suficientemente compleja como para intentar buscar nuevos caminos que ofrezcan la materia de un modo mas natural y atractivo.
- un elevado componente práctico basado en la aplicación de casos reales favorece la asimilación de los conceptos teóricos de la disciplina.
- la motivación del alumno es un parámetro fundamental que se ve incrementado mediante la experimentación “in situ” en empresas durante determinados períodos de presencia física.
- la incorporación del uso de las nuevas tecnologías abre prometedoras perspectivas de comunicación e intercambio de información entre alumnos-profesores. Ello facilita las labores de asesoramiento y seguimiento en la formación académica.

10. BIBLIOGRAFIA

- [1] Booch G. *Object-Oriented Analysis and Design with Applications*, Second Edition. Benjamin/Cummings, 1994.
- [2] Benedito, V. *Introducción a la didáctica*. Barcanova 1987.
- [3] Castillejo, J.L. *Pedagogía tecnológica*. Ceac, 1987.
- [4] CSI. Metodología de Planificación y Desarrollo de Sistemas de Información Métrica versión 2, Guía de Referencia: Serie Administración General, 1993.
- [5] CSI. Metodología de Planificación y Desarrollo de Sistemas de Información Métrica versión 2, Guía Técnicas: Serie Administración General, 1993.
- [6] Pages-Jones. *The practical guide to structured system design*. M. Yourdon Press 1980.
- [7] Roger S. Pressman. *Ingeniería del sw. Un enfoque práctico (3ª ed.)*. Mc Graw-Hill 1993.
- [8] Senn, J.A. . *Análisis y Diseño de Sistemas de Información*, Mac Graw-Hill, 1992.
- [9] Yourdon E. *Modern Structured Analysis*. Prentice-Hall 1989.